

Fakta

Kurs:	Systemutvecklare C/C++ Extended 2024
Klass:	SUVx24
Teknikområde:	Programspråket C, in- och utmatning, funktioner

Learning Target

Programmering i C och C++

Grundläggande begrepp inom programmering.

Skapa källkod med C syntax. Tema funktioner.

Grundläggande in- och utmatning

In- och utmatning av resultat från ett program med hjälp av standardfunktioner i C.

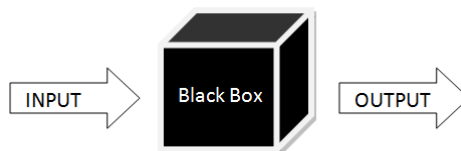
Användning av utvecklingsmiljöer

Använda *Visual Studio Code* (förkortat VSC) för att skapa källkod, kompilera och exekvera program skapat med programspråket C.

Introduktion

Att skriva program som en lösning på ett problem, leder ofta till många rader kod. Ett sätt att korta ner mängden kod i ett avsnitt, är att bryta ut den och lägga blocket i en separat del. Detta kan lösas med hjälp av funktioner i C.

En annan vanlig användning av funktioner är att se den som en svart låda. Ge lådan ett stimuli i form av indata (input), låt lådan (funktionen) bearbeta indata, lådan ger sedan feedback genom att returnera utdata (output) från lådan. Detta mönster används flitigt i programspråk.



Om vi använder en funktion som inte returnerar något tillbaka, kan vi i stället kalla det för en procedur. Den bara utför något, men ska inte lämna tillbaka något som resultat av bearbetningen. Dock kan proceduren mycket väl utföra visuell feedback som output. Detta sätt att använda funktion gör det möjligt att partitionera koden i lämpliga delar.

Workshop

Förberedelse

Skapa en ny projektmapp (katalog) i den projektmapp du använder för kodning, typ:

```
C:\Chas\SUVx24\Workshop10\
```

Starta VSC på det sätt du tycker är lättast och öppna den nya projektmappen.

Innehåll

Träna på att använda funktioner och göra refaktorering.

Uppgift 1a

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-1a.c**

Skapa ett program som

- Har en funktion `printname()` som skriver ut ditt fullständiga namn.
- Anropa `printname()` tre gånger i huvudfunktionen `main()`.
- Glöm inte att göra en funktionsdeklaration överst i programmet.

Extra: Gör en inmatning för hur många gånger programmet ska skriva ut namnet. Anropa funktionen så många gånger. **Tips:** Du kan kopiera koden från **workshop7-1-extra.c** och refaktorerar den.

Uppgift 1b

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-1b.c**

Kopiera och refaktorerar **workshop10-1a.c**.

- Ta bort deklarationen av funktionen och flytta definitioner före `main()`-funktionen.
- Varför klarar vi oss utan deklaration?
- Var kan den hamna i stället?
- Vad är 'best practice'?

Workshop

Uppgift 2

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-2.c**

Skapa ett program som

- Har en funktion `printline()` som skriver ut 80 st bindestreck i en följd, som en linje.
- Använd funktionen `printline()` för att skapa följande utseende med ett huvudprogram som anropar funktionen på lämpliga ställen:

```
-----  
C Programming Language Author(s)  
-----  
1. Brian W. Kernighan  
2. Dennis M. Ritchie  
-----
```

- Använd både deklaration och definition av funktionen.

Uppgift 3a

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-3a.c**

Kopiera koden från **workshop10-2.c** och refaktorera den genom att

- Låta funktionen `printline()` ha en parameter `char symbol` som kan ta emot ett argument med vilken symbol som helst i stället för bindestreck varje gång.
- Modifiera huvudprogrammet att ge följande utskrift:

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
C Programming Language Author(s)  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
1. Brian W. Kernighan  
2. Dennis M. Ritchie  
*****
```

Workshop

Uppgift 3b

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-3b.c**

Kopiera koden från **workshop10-2.c** och refaktorera den genom att

- Låta funktionen `printline()` ha ytterligare en parameter `int n` som talar om hur många tecken som ska skrivas ut på en rad.
- Modifiera funktionen att använda `int n` och skriva ut rätt antal tecken.
- Modifiera huvudprogrammet att ge följande utskrift:

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
                        C Programming Language Author(s)
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
                        1. Brian W. Kernighan
                        2. Dennis M. Ritchie
*****
```

Uppgift 4

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop10-4.c**

Skapa ett program som

- Har en funktion som alltid returnerar värdet 143.
- Gör ett huvudprogram som har en variabel `n` initierad till 0.
- Gör en utskrift av värdet före anrop.
- Tilldela resultatet från funktionen med anrop till den.
- Skriv ut värdet efter anropet.